項グラフ言語の正データからの多項式時間帰納推論可能性について(計算理論とその応用)

林, 夕起子; 松本, 哲志; 正代, 隆義

数理解析研究所講究録
992巻 1997年 66-73

項グラフ言語の正データからの多項式時間帰納推論可能性について

林 夕起子　　松本 哲志*　　正代 隆義
九州大学大学院システム情報科学研究科情報理学専攻
{hayashi, matumoto, shoudai}@i.kyushu-u.ac.jp

1. はじめに

トポロジカルな構造をもった代表的な例としてグラフがある．グラフは，現実社会における様々なデータを表現する1つの手段として重要な役割を果たしている．実際，分子生物学において，RNAの2次構造は木で表せることが知られている．さらに，グラフ文法やグラフ言語は，広範囲にわたって研究されており，パターン認識，ソフトウェアの仕様や開発，データベース，VLSI設計など様々な応用が試みられている．本論文では，グラフで表現されたいくつかのデータから，それらに共通したある特徴を持つグラフを見つける問題について考察する．ここで，共通したある特徴をもつグラフとして超グラフ [2] を基にして内田ら [7, 8] によって導入された項グラフを用いる．さらに，そのようなグラフを見つけることは本質的に計算機にできることなのか，それは妥当な時間で見つけることは可能であるかについて「極限における同定」という枠組み [1, 3] を用いて論じる．特に，本論文では正データからの多項式時間帰納推論について論じる．正データから多項式時間帰納推論では，所属性問題と極小言語問題が重要な鍵となる．所属性問題とは，ある例とある規則が与えられたとき例が規則に属するかという問題であり，極小言語問題とは，空でない例の集合が与えられたとき，それらの例をうまく説明している極小な規則を見つける問題である．

本論文では，最初に木を生成する項グラフの一種として正則項線形連鎖と正則項 $(*,1)$- キャタピラを定義する．一般に，キャタピラとは線形連鎖の各頂点に対していくつかの線形連鎖が伸びている木をいう．次に，正則項線形連鎖と正則項 $(*,1)$- キャタピラの概念クラスに対する極小言語問題を解く多項式時間アルゴリズムが存在するための十分条件を示す．さらに，ある表現形式を用いることにより，その十分条件を満たすことを示す．最後に，これらの概念クラスに対する極小言語問題を解く多項式時間アルゴリズムを提案し，多項式時間で推論可能であることを示す．

2. 準備

この節では，グラフに関する諸定義と正データからの帰納推論に関する諸定義について述べ，これまでに得られている結果についてまとめる．

2.1. 項グラフ

$g = (V, E)$ を無向グラフとする．特に，$E = \emptyset$ であるグラフを空グラフとよぶ．Σ, Λ を有限アルファベットとするとき，関数 $\varphi : V \rightarrow \Sigma$ を頂点のラベル付け関数とよび，関数 $\psi : E \rightarrow \Lambda$ を辺のラベル付け関数とよぶ．このとき，4つ組 $g = (V, E, \varphi, \psi)$ を (Σ, Λ) 上の彩色グラフという．A を集合としたとき，その要素数を $|A|$ で表す．X を可算集合とし，その要素を変数とよび，$x, y, \cdots$ で表す．$\Sigma \cap X = \emptyset, \Lambda \cap X = \emptyset$ とし，それぞれの変数 $x \in X$ は，ランクと呼ばれる非負整数をもつとする．

*日本学術振興会特別研究員

定義 1. [8, 7] **項グラフ**とは，$\langle\Sigma,\Lambda,X\rangle$ 上の7つ組 $g=(V,E,\varphi,\psi,H,\lambda,P)$ によって定義される (図1参照). 各成分は次のようなものである.

1. (V,E,φ,ψ) は (Σ,Λ) 上の彩色グラフである.
2. H は V の巾集合 2^V の部分集合である. その要素を超辺とよぶ.
3. 関数 $\lambda: H\to X$ を超辺のラベル付け関数とよぶ. 但し，超辺 $e\in H$ にラベル付けされた変数のランクは $|e|$ と等しいものとする. これを e のランクとよぶ.
4. $P=\{p(e)\}_{e\in H}$ とする. ここで, $p(e)$ は全単射 $p(e):e\to\{1,2,\cdots,|e|\}$ である. すなわち，$p(e)$ は e の要素に対して，それぞれ異なる番号を割り当てる関数とする. e の要素をポートとよぶ.

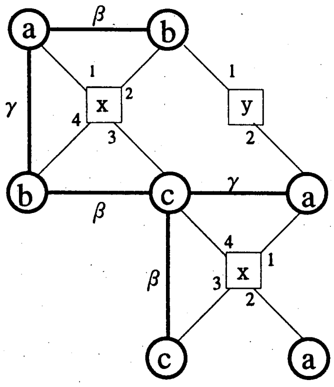

図 1: 項グラフ

定義 2. [8, 7] (Σ,Λ,X) 上の**正則項グラフ**とは，全ての $x\in X$ に対して，x でラベル付けされた超辺が高々1つしか存在しない項グラフである. すなわち，超辺のラベルがそれぞれ相異なる項グラフである.

文脈から明らかな場合は，(Σ,Λ,X) を省略する. 項グラフは $f,g,...$ で表す. 特に，$H=\emptyset$ である項グラフを**基礎項グラフ**とよぶ. 基礎項グラフは，彩色グラフと同一視できる. ここで，項グラフに対するいくつかの定義を与える.

$f=(V_f,E_f,\varphi_f,\psi_f,H_f,\lambda_f,P_f)$ ，$g=(V_g,E_g,\varphi_g,\psi_g,H_g,\lambda_g,P_g)$ を項グラフとする. 次の条件を満たすとき，f と g は同型であるといい，$f\simeq g$ で表す.

(1) (Σ,Λ) 上の彩色グラフ $(V_f,E_f,\varphi_f,\psi_f)$ と $(V_g,E_g,\varphi_g,\psi_g)$ は，同型である.

(2) $\phi: V_f\to V_g$ を (1) で与えた全単射とする. そのとき，全単射 $\varpi: H_f\to H_g$ が存在し，任意の $e\in H_f$ に対して，$\varpi(e)=\phi^*(e)$，$p_f(e)=p_g(\varpi(e))$，$\lambda_f(e)=\lambda_g(\varpi(e))$ となる. ここで $\phi^*(\{v_1,\cdots,v_m\})=\{\phi(v_1),\cdots,\phi(v_m)\}$ とする.

$f=(V_f,E_f,\varphi_f,\psi_f,H_f,\lambda_f,P_f)$，$g=(V_g,E_g,\varphi_g,\psi_g,H_g,\lambda_g,P_g)$ を項グラフとする. ここで，$\sigma=(v_1,v_2,\cdots,v_r)$ を g の r 個の相異なる頂点の列とする. このとき2つ組 $[g,\sigma]$ を g の **r- 超グラフ**とよぶ. ここで, ランク e の超辺 $e=\{u_1,u_2,\cdots,u_r\}\in H_f$ に対して, 関数 $p_f(e)$ によって $u_1,u_2,\cdots,u_r$ の順で番号が割り当てられているとする. このとき，f 上の**超辺置き換え** $e\leftarrow[g,\sigma]$ とは，f に g を組み込む手続きで，次のように定義する：

1. 超辺 $e\in H_f$ は，関数 $p_f(e)$ によって $u_1,u_2,\cdots,u_r$ の順に番号が割り当てられているので，その番号と $\sigma=(v_1,v_2,\cdots,v_r)$ の順に従って，u_i に v_i $(1\le i\le r)$ を重ね合わせる.
2. 1 によって重ね合わされた頂点のラベルは，u_i $(1\le i\le r)$ のラベルを優先する.

このように f 上の超辺置き換え $e\leftarrow[g,\sigma]$ によって得られた項グラフを $f(e\leftarrow[g,\sigma])$ で表す. $\Upsilon=\{e_1\leftarrow[g_1,\sigma_1],...,e_m\leftarrow[g_m,\sigma_m]\}$ を f 上の超辺置き換えの集合とする. そのとき，Υ のすべての超辺置き換えを適用して得られる項グラフを $f(\Upsilon)$ で表す. $f(\Upsilon)$ は超辺置き換えを行なう順番には依存しない. 変数 x でラベル付けられているすべての超辺 e に対して，超辺置き換え $e\leftarrow[g,\sigma]$ を行なうことを $x:=[g,\sigma]$ で表し，x の**束縛**とよぶ.

定義 3. [8, 7] **代入** θ とは，束縛の集合 $\{x_1:=[g_1,\sigma_1],\cdots,x_n:=[g_n,\sigma_n]\}$ である (図2 参照). ただし，x_i $(1\le i\le n)$ は，互いに異なる変数である.

$f=(V_f,E_f,\varphi_f,\psi_f,H_f,\lambda_f,P_f)$，$g=(V_g,E_g,\varphi_g,\psi_g,H_g,\lambda_g,P_g)$ を項グラフとし，$\theta=\{x_1:=[g_1,\sigma_1],\cdots,x_n:=[g_n,\sigma_n]\}$ を代入とする. 変数 x_i に対して，$H_{\lambda_f}(x_i)=\{e\in H_f\mid\lambda_f(e)=x_i\}$，$\Upsilon_i=\{e\leftarrow[g_i,\sigma_i]\mid e\in H_{\lambda_f}(x_i)\}$ とし，$\Upsilon_\theta=\bigcup_{i=1}^{n}\Upsilon_i$ とする.

このとき，項グラフ $f\theta$ を θ による f の代入例とよび，$f(\Upsilon_\theta)$ によって定義する．$f\theta$ の例を図2に示す．ここで，代入は，$\theta = \{x := [g_1, (v_1, v_3)],\ y := [g_2, (u_1, u_2, u_3)]\}$ とする．任意の $e_i, e_j \in H_f\ (i \neq j)$ に対して，$e_i \not\subseteq e_j$ を満たすとき，f は標準形であるとよぶ．ある代入 θ, θ' に対して，$f \simeq g\theta$ かつ $g\theta' \simeq f$ が成り立つならば，f は g の変種とよぶ．$f \simeq g\theta$ を満たす代入 θ が存在するとき，g は f より一般的である，または f は g より具体的であるといい，$f \preceq g$ で表す．ここで扱う項グラフ全体の集合を $\mathcal{TG}$ で表す．ただし，任意の $g \in \mathcal{TG}$ は標準形であり，任意の $f, g \in \mathcal{TG}$ に対して，f は g の変種ではないものとする．また，基礎項グラフ全体の集合を $\mathcal{GTG}$ で表す．

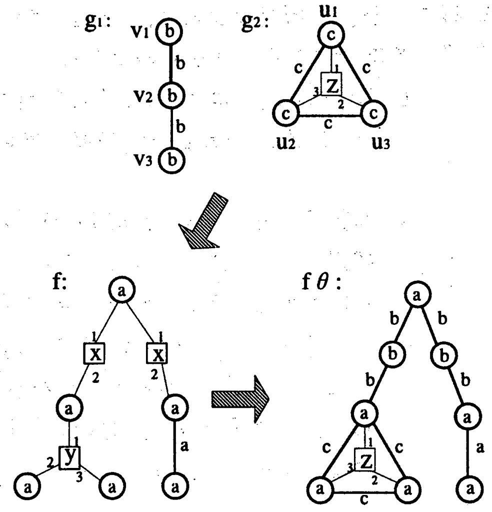

図 2: 代入例 $f\theta$

定義 4. g を項グラフとする．g の**項グラフ言語**とは，g より具体的な基礎項グラフ全体の集合で，$L(g)$ で表す．すなわち，$L(g) = \{w \in \mathcal{GTG} \mid w \preceq g\}$ である．特に，g が正則項グラフであるとき，$L(g)$ を**正則項グラフ言語**と呼ぶ．項グラフ言語全体の族を $\mathcal{L}(\mathcal{TG})$ で表し，正則項グラフ言語全体の族を $\mathcal{L}(\mathcal{RTG})$ で表す．

2.2. 正データからの帰納推論

帰納推論 [3] とは，与えられたデータからそれを説明する一般的な規則を推測する過程である．U を帰納的に可算な集合とし，**普遍的な集合**とよぶ．U の要素を**語**とよび，U の部分集合 L を**概念**とよぶ．本稿では，U を基礎項グラフ全体の集合 $\mathcal{GTG}$ とし，その部分集合を**言語**と呼ぶ．語 w の長さを $|w|$ と書く．

定義 5. **帰納的概念の添字付き族** または**概念クラス**とは，3つ組 $\mathcal{C} = (C, R, r(\cdot))$ によって定義される．各成分は次のようなものである．

1. C は概念の族である．

2. R を帰納的可算集合であり，R の要素を**表現**と呼ぶ．

3. 写像 $r : R \to C$ は全射である．

4. 次のような帰納的関数 $f : U \times R \to \{0,1\}$ が存在する．

$$f(w,g) = \begin{cases} 1, & w \in r(g)\text{のとき}, \\ 0, & \text{そうでないとき}. \end{cases}$$

文脈から明らかな場合は，$r(\cdot)$ を除き，2つ組 $\mathcal{C} = (C, R)$ で表すことにする．

定義 6. $\mathcal{C} = (C, R, r(\cdot))$ を概念クラスとする．空でない概念 L の**正提示**とは，$L = \{w_1, w_2, \cdots\}$ を満たす U の要素の無限列 $w_1, w_2, \cdots$ である．

$\mathcal{C} = (C, R, r(\cdot))$ を概念クラスとする．$\mathcal{C}$ に対する**推論機械** M とは，ときどき入力を要求し，ときどき R の要素を出力するような手続きである．推論機械によって出力された表現を**仮説**と呼ぶ．M に対して，$w_1, w_2, \cdots$ の順に入力が与えられたとき，M によって i $(i = 1, 2, \cdots)$ 番目に生成される仮説を h_i とする．このとき，M が**反応的** であるとは，任意の i に対して，M が入力として w_i を受け取り，次の入力 w_{i+1} を受け取る前に仮説 h_i を出力することをいう．M が**無矛盾**であるとは，任意の i に

対して，$r(h_i) \supseteq \{w_1, \cdots, w_i\}$ であることをいう．M が保守的であるとは，$h_{i-1} \neq h_i$ であるならば，$w_i \notin r(h_{i-1})$ であることをいう．M が正データから概念クラス $\mathcal{C}$ に関して概念 L を極限において同定するとは，L の任意の正提示を M に与えたとき，M の生成する仮説の無限列が $L = r(h)$ となる h に収束することをいう．M が正データから概念クラス $\mathcal{C}$ を推論するとは，任意の概念 $L \in C$ に対して，M が正データから概念 L を極限において同定することをいう．概念クラス $\mathcal{C}$ が正データから推論可能であるとは，$\mathcal{C}$ を正データから推論する推論機械 M が存在することをいう．$\mathcal{C} = (C, R, r(\cdot))$ を概念クラスとする．$\mathcal{C}$ が正データから多項式時間推論可能であるとは，正データから $\mathcal{C}$ を推論する無矛盾でかつ保守的で反応的な推論機械 M が存在し，任意の i に対して，M が仮説 h_i を生成するまでの時間が今までにもらった入力の長さの和 $|w_1| + |w_2| + \cdots + |w_i|$ の多項式時間であることをいう．

定義 7. 概念クラス $\mathcal{C} = (C, R)$ が有限の厚みをもつとは，任意の $w \in U$ に対して $|\{L \in C \mid w \in L\}|$ が有限であることをいう．

Angluin [1] は，次のような正データからの推論可能性の十分条件を示している．

定理 1. [1] 概念クラス $\mathcal{C} = (C, R)$ が有限の厚みをもつならば，$\mathcal{C}$ は正データから推論可能である．

補題 1. $\mathcal{C}_{\mathcal{TG}} = (\mathcal{L}(\mathcal{TG}), \mathcal{TG})$ とする．このとき，$\mathcal{C}_{\mathcal{TG}}$ は有限の厚みをもつ．

定理 1と補題 1より，以下の定理が成り立つ．

定理 2. $\mathcal{C}_{\mathcal{TG}}$ は正データから推論可能である．

帰納推論の現実的な問題を考える場合，推論可能が保証されただけでは不十分であり，効率のよい推論が重要となる．そこで，概念クラス $\mathcal{C}$ の多項式時間推論可能性について考察するために Angluin [1] は，$\mathcal{C}$ に対する極小言語問題を定義した．

定義 8. 概念クラス $\mathcal{C}$ に対する極小言語問題

入力： 空でない有限集合 $S \subseteq U$．

問題： $S \subseteq r(g)$ であり任意の $h \in R$ に対して，$r(h) \subsetneq r(g)$ ならば $S \not\subseteq r(h)$ となる $g \in R$ を見つけよ．

概念クラス $\mathcal{C} = (C, R)$ に対する極小問題を解くアルゴリズムを *MINL* とする．このとき，Angluin は次のアルゴリズム *INFER* (図 3 参照) を用いて，$\mathcal{C}$ を正データから無矛盾で保守的でかつ反応的に推論するための十分条件を得た．

補題 2. [1] 概念クラス $\mathcal{C} = (C, R)$ が有限の厚みをもち，$\mathcal{C}$ に対する極小言語問題が多項式時間計算可能ならば，アルゴリズム *INFER* は正データから無矛盾でかつ保守的で反応的に $\mathcal{C}$ を多項式時間で推論する．

篠原 [6] は，多項式時間推論可能性について考察するために Angluin [1] のアルゴリズム *INFER*を使って $\mathcal{C}$ が多項式時間推論可能であるための補題を得た．

定義 9. 概念クラス $\mathcal{C}$ に対する所属性問題

入力： 語 $w \in U$, 表現 $g \in R$.

問題： w は $r(g)$ に属するか？

補題 3. [6] 概念クラス $\mathcal{C}$ が有限の厚みをもち，$\mathcal{C}$ に対する所属性問題と極小言語問題が多項式時間計算可能ならば，$\mathcal{C}$ は正データから多項式時間推論可能である．

```
Procedure INFER
input:  無限列 w1, w2, ··· ∈ U.
output: 無限列 h1, h2, ··· ∈ R.
begin
  S := φ;                       /* 今までに受け取った例の集合 */
  h := "none";                  /* 仮説 */
  for i := 1 to ∞ do
    begin
      S := S ∪ {wi};            /* 次の例を受け取る */
      if S ⊄ r(h) then          /* 無矛盾であるかチェックする */
        h := MINL(S);           /* S を含む極小な概念の表現を見つける */
    output(h)
    end
end.
```

図 3: Procedure *INFER*

3. 項木言語の正データからの多項式時間推論可能性

この節では，言語族 $\mathcal{C}_{\mathcal{TG}} = (\mathcal{L}(\mathcal{TG}), \mathcal{TG})$ に対する多項式時間推論可能性について論じる．定理 2より，$\mathcal{C}_{\mathcal{TG}}$ は正データから推論可能であるので，$\mathcal{C}_{\mathcal{TG}}$ が正データから多項式時間推論可能であるか考察するために，$\mathcal{C}_{\mathcal{TG}}$ に対する次の 2 つの問題について考える．

定義 10. $\mathcal{C}_{\mathcal{TG}}$ に対する所属性問題

入力： 基礎項グラフ $w \in \mathcal{GTG}$， 項グラフ $g \in \mathcal{TG}$.

問題： w は L_g に属するか？ すなわち，$w = g\theta$ を満たす代入 θ が存在するか？

定義 11. $\mathcal{C}_{\mathcal{TG}}$ に対する極小言語問題

入力： 空でない基礎項グラフの有限集合 S .

問題： $S \subseteq L(g)$ であり任意の $r \in \mathcal{TG}$ に対して，$L(h) \subsetneq L(g)$ ならば $S \not\subseteq L(h)$ となる 項グラフ $g \in \mathcal{TG}$ を見つけよ．

部分グラフ同型問題が NP 完全であることより，$\mathcal{C}_{\mathcal{TG}}$ に対する所属性問題は NP 完全である．このことより，P≠NP の仮定の下では Angluin [1] 及び篠原 [6] のアルゴリズムでは，多項式時間で推論できないことが分かる．そこで，我々は次のような $\mathcal{L}(\mathcal{TG})$ の部分族を定義し，この言語族に対する所属性問題について考察した．

定義 12. 基礎項グラフが木であるとき**基礎項木**と呼ぶ．基礎項木全体の集合を $\mathcal{GTT}$ で表す．

定義 13. g を項グラフとし，$x_1, x_2, \cdots, x_n$ を g に現れる全ての超辺のラベルとする．$g_1, g_2, \cdots, g_n$ を任意の基礎項木とするとき，代入 $\theta = \{x_1 := [g_1, \sigma_1], x_2 := [g_2, \sigma_2], \cdots, x_n := [g_n, \sigma_n]\}$ によって得られる基礎項グラフ $g\theta$ が基礎項木であるとき，g を**項木**と呼ぶ．特に，項木の超辺のラベルがそれぞれ異なるとき，**正則項木**と呼ぶ．

定義 14. 項木 t の**項木言語**とは，t に任意の基礎項木を代入してできる基礎項グラフ全体の集合で $L_T(t)$ で表す. 特に，t が正則項木であるとき $L_T(t)$ を**正則項木言語**と呼ぶ.

正則項木全体の集合を $\mathcal{RT}$ ，正則項木言語全体の族を $\mathcal{L}(\mathcal{RT})$ で表す. ここで，$\mathcal{C}_{\mathcal{RT}} = (\mathcal{L}(\mathcal{RT}), \mathcal{RT})$ に対する所属性問題を次のように定義する.

定義 15. $\mathcal{C}_{\mathcal{RT}}$ に対する所属性問題

入力： 基礎項木 $w \in \mathcal{GTT}$, 正則項木 $t \in \mathcal{RT}$

問題： w は $L_T(t)$ に属するか?

部分木同型問題が多項式時間で計算可能 [5] であることより，次の命題が成り立つ.

命題 1. $\mathcal{C}_{\mathcal{RT}}$ に対する所属性問題は多項式時間計算可能である.

3.1. 極小言語問題

次に $\mathcal{C}_{\mathcal{TT}}$ に対する極小言語問題について考察する. 2 つの項木が与えられたとき，それらの間の同型写像が複数存在する. このことは，極小言語問題を解くことを難しくしている. よって，我々は辺とポート数が 2 の超辺からなる正則項木にのみ着目し，さらに頂点のラベルは考えないものとする. また，このような項木全体の集合を $\mathcal{RT}^2$ で表す. この節では，$\mathcal{RT}^2$ の部分集合である正則項 (k,l)− キャタピラを定義し，その概念クラスに対する極小言語問題について考察する.

項木 $f = (V_f, E_f, \varphi_f, \psi_f, H_f, \lambda_f, P_f) \in \mathcal{RT}^2$ に対して，f の頂点 v の**次数** (*degree*) とは，v を端点としてもつ辺と超辺の数の和であり，$deg(v)$ で表す. すなわち，$deg(v) = |\{e \in E_f \mid v \in e\}| + |\{h \in H_f \mid v \in h\}|$ である. $f, g \in \mathcal{RT}$ に対して，f と g のすべての超辺を辺に置き換えることによって得られる基礎項木が，辺のラベルを除いて同型であるならば $f \approx g$ と表す. ここで，$\mathcal{RT}^2$ の部分集合である項線形連鎖と項キャタピラについて定義する.

定義 16. **項線形連鎖** $g = (V, E, \varphi, \psi, H, \lambda, P)$ とは，$V = \{v_1, v_2, v_3, \cdots, v_n\}$ としたとき，$E \cap H = \emptyset$ かつ $E \cup H = \{\{v_1, v_2\}, \{v_2, v_3\}, \cdots, \{v_{n-1}, v_n\}\}$ である項木である (図 4参照). 特に，項線形連鎖の超辺のラベルがそれぞれ異なるとき，**正則項線形連鎖**という.

項線形連鎖 g のサイズを g の $E \cup H$ の個数で定義する. $|E \cup H| = n$ であるとき，これをサイズ n の項線形連鎖と呼ぶ.

定義 17. [4] キャタピラとは，背骨と呼ばれる線形連鎖の各頂点に対して，いくつかの線形連鎖が伸びている木である. この伸びている線形連鎖をヘアーと呼ぶ.

定義 18. **項キャタピラ**とは，背骨やヘアーが項線形連鎖であるようなキャタピラである. 項キャタピラの超辺のラベルがそれぞれ異なるとき，**正則項キャタピラ**とよぶ. 整数 $k, l \geq 0$ に対して**項 (k,l)- キャタピラ** とは，各頂点に対するヘアーの数が高々 k で，ヘアーの最大のサイズが高々 l であるような項キャタピラである. 項 (k,l)- キャタピラの超辺のラベルがそれぞれ異なるとき，**正則項 (k,l)- キャタピラ**とよぶ (図 5 参照).

定義 19. 項線形連鎖 f の**項線形連鎖言語** とは，f に基礎項木を代入してできる基礎項グラフ全体の集合で，$L_T(f)$ で表す. f が正則項線形連鎖であるとき，**正則項線形連鎖言語** とよぶ.

定義 20. 項 (k,l)- キャタピラ g の**項 (k,l)- キャタピラ言語** とは，g に基礎項木を代入してできる基礎項グラフ全体の集合で $L_T(g)$ で表す. g が正則項 (k,l)- キャタピラであるとき，**正則項 (k,l)- キャタピラ言語**と呼ぶ.

正則項線形連鎖全体の集合を $\mathcal{RT}_{linear}$ ，正則項 (k,l)- キャタピラ全体の集合を $\mathcal{RT}_{(k,l)-cat}$ で表し，正則項線形連鎖言語全体の族を $\mathcal{L}(\mathcal{RT}_{linear})$，正則項 (k,l)- キャタピラ言語全体の族を $\mathcal{L}(\mathcal{RT}_{(k,l)-cat})$ で表す．さらに，$\mathcal{RT}_{(*,l)-cat} = \bigcup_{k\geq 0}\mathcal{RT}_{(k,l)-cat}$ とする．

概念クラス $\mathcal{C}_{\mathcal{RT}_{(*,l)-cat}}$=$(\mathcal{L}(\mathcal{RT}_{(*,l)-cat}),\mathcal{RT}_{(*,l)-cat})$ とする．次に，$\mathcal{C}_{\mathcal{RT}_{(*,l)-cat}}$ に対する極小言語問題について考察する．

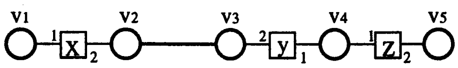

図 4: 長さ 4 の正則項線形連鎖

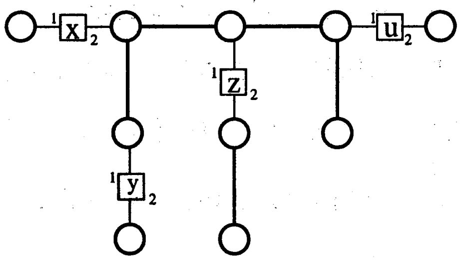

図 5: 正則項 (1,2)- キャタピラ

次に，極小言語問題を解くときに有用な条件を定義する．$\mathcal{D} \subseteq \mathcal{RT}^2$ がこの条件を満たせば，包含関係に関して極小なものを見つける問題を関係 $\preceq$ に関して極小なものを見つける問題に置き換えることができる．

定義 21. $\mathcal{D} \subseteq \mathcal{RT}^2$ とする．$\mathcal{D}$ が条件 1 を満たすとは，任意の $f, g \in \mathcal{D}$ に対して $f \approx g$ であるとき $L_T(f) \subseteq L_T(g)$ であることと $f \preceq g$ であることが同値であるときをいう．

例 1. $|\Lambda| = 1$ とする．$\mathcal{RT}_{linear}$ は条件 1 を満たさない．

例 2. $|\Lambda| = 2$ とする．このとき，$\mathcal{RT}_{(1,2)-cat}$ は条件 1 を満たさない．

補題 4. $|\Lambda| = 2$ とする．このとき，$\mathcal{RT}_{(1,1)-cat}$ は条件 1 を満たす．

命題 1，補題 4より次の定理が成り立つ．

定理 3. 概念クラス $\mathcal{C}_{(1,1)-cat}$ は正データから多項式時間推論可能である．

$|\Lambda| = 1$ の正則項線形連鎖に関して，例 1より条件 1 は満たさない．そこで，次のような $\mathcal{D} \subseteq \mathcal{RT}^2$ の部分族 $\mathcal{D}' \subseteq \mathcal{RT}^2$ を定義する．

定義 22. $\mathcal{D} \subseteq \mathcal{RT}^2$ とする．任意の項木 $g = (V, E, \varphi, \psi, H, \lambda, P) \in \mathcal{D}$ を項木とする．このとき，$\mathcal{D}' \subseteq \mathcal{D}$ を次の 1, 2 を満たす項木全体とする．

1. u_1 を葉とする辺または超辺 $\{u_1, u_2\} \in E \cup H$ とその隣接辺または隣接超辺 $\{u_2, u_3\}$ に対し，$deg(u_2) = 2$ かつ $\{u_2, u_3\}$ が超辺であるならば $\{u_1, u_2\}$ も超辺である．
2. 3 つの連続する $E \cup H$ の要素を $\{v_1, v_2\}$，$\{v_2, v_3\}$，$\{v_3, v_4\}$ とする．このとき，$deg(v_2) = 2$ または $deg(v_3) = 2$ であり，かつ $\{v_1, v_2\}$，$\{v_3, v_4\}$ のどちらも超辺であるならば $\{v_2, v_3\}$ も超辺である．

ここで，定義 22を満たす $\mathcal{RT}'_{(*,1)-cat}$ から得られる言語全体の族を $\mathcal{L}(\mathcal{RT}'_{(*,1)-cat})$ で表す．
すなわち，$\mathcal{L}(\mathcal{RT}'_{(*,1)-cat}) = \{L_T(g) \mid g \in \mathcal{RT}'_{(*,1)-cat}\}$ である．このとき，$\mathcal{L}(\mathcal{RT}'_{(*,1)-cat})$ に対して，以下のことがいえる．

補題 5. $\mathcal{L}(\mathcal{RT}_{(*,1)-cat}) = \mathcal{L}(\mathcal{RT}'_{(*,1)-cat})$.

補題 6. $\mathcal{RT}'_{(*,1)-cat}$ は条件 1 を満たす．

命題 1，補題 5，補題 6より次の定理が成り立つ．

定理 4. 概念クラス $\mathcal{C}_{(*,1)-cat}$ は正データから多項式時間推論可能である．

4. まとめ

本論文では，木を生成する正則項 (k,l)- キャタピラを定義し，その言語のクラスの正データからの多項式時間推論可能性について，辺のラベルの数が 1 の場合と 2 の場合に分けて論じた．最初に，ある項グラフの概念クラス $\mathcal{C}$ が，極小言語問題を解く多項式時間アルゴリズムを持つ十分条件について述べた．正則項 $(*,1)$- キャタピラの概念クラス $\mathcal{C}_{\mathcal{RT}_{(*,1)-cat}}$ は，辺のラベルの数が 1 の場合，この十分条件をみたさないが，新たな表現形式を用いることにより正データからの多項式時間推論可能であることを示した．次に，辺のラベルの数が 2 の場合は，新たな表現形式を用いずに正則項 $(1,1)$- キャタピラの概念クラス $\mathcal{C}_{\mathcal{RT}_{(1,1)-cat}}$ が正データから多項式時間推論可能であることを示した．

正則項 (k,l)- キャタピラの概念クラス $\mathcal{C}_{\mathcal{RT}_{(k,l)-cat}}$ や正則項木の概念クラス $\mathcal{C}_{\mathcal{RT}}$ が正データから多項式時間推論可能であるかどうかは今後の課題である．また，ポート数 3 以上の超辺を含む項グラフに関しても未解決である．グラフの正データからの多項式時間推論可能性を考えるにあたって，2 つのグラフが与えられたとき，それらの間に複数の同型写像が存在することがこの問題を難しくしている．

辺のラベル	仮説空間	仮説の表現	条件 1	多項式時間帰納推論可能
$\|\Lambda\| = 1$	$\mathcal{L}(\mathcal{RT}_{linear})$	RT_{linear}	No	Yes
		RT'_{linear}	Yes	
	$\mathcal{L}(\mathcal{RT}_{(*,1)-cat})$	$RT_{(*,1)-cat}$	No	Yes
		$RT'_{(*,1)-cat}$	Yes	
	$\mathcal{L}(\mathcal{RT}_{(1,2)-cat})$	$RT_{(1,2)-cat}$	No	未解決
$\|\Lambda\| = 2$	$\mathcal{L}(\mathcal{RT}_{linear})$	RT_{linear}	Yes	Yes
	$\mathcal{L}(\mathcal{RT}_{(1,1)-cat})$	$RT_{(1,1)-cat}$	Yes	Yes
	$\mathcal{L}(\mathcal{RT}_{(1,2)-cat})$	$RT_{(1,2)-cat}$	No	未解決

表 1: まとめ

参考文献

[1] D. Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.

[2] C. Berge. *Hypergraphs.* North-Holland, 1989.

[3] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[4] O. Maruyama and S. Miyano. Graph inference from a walk for trees of bounded degree 3 is NP-complete. In *Mathematical Foundations of Computer Science 1995 (Lecture Notes in Computer Science 969)*, pages 257–266, 1995.

[5] S. W. Reyner. An analysis of a good algorithm for the subtree problem. *SIAM Journal of Computation*, 6:730–732, 1977.

[6] T. Shinohara. *Studies on Inductive Inference from Positive Data.* PhD thesis, Kyushu University, 1986.

[7] T. Uchida. *Formal Graph Systems and Parallel Graph Algorithm Design.* PhD thesis, Kyushu University, 1994.

[8] T. Uchida, T. Shoudai and S. Miyano. Parallel algorithm for refutation tree problem on formal graph systems. *IEICE Transactions on Information and Systems*, E78-D:99–112, 1995.